On the evaluation of finite-time ruin probabilities in a dependent risk model

Dimitrina S. Dimitrova*, Vladimir K. Kaishev and Shouqi Zhao

Cass Business School, City University London, UK

Abstract

This paper establishes some enlightening connections between the explicit formulas of the finite-time ruin probability obtained by Ignatov and Kaishev (2000, 2004) and Ignatov et al. (2001) for a risk model allowing dependence. The numerical properties of these formulas are investigated and efficient algorithms for computing ruin probability with prescribed accuracy are presented. Extensive numerical comparisons and examples are provided.

Keywords: finite time ruin probability, dependent risk modelling, Appell polynomials, numerical implementation, order statistics

1 Introduction

Research on ruin probability beyond the classical risk model has intensified in recent years. More general ruin probability models assuming dependence between either claim amounts or claim arrivals, or cross-dependence between both arrivals and sizes of claims, and non-linear aggregate premium income have been considered in the actuarial and applied probability literature. Such models are better suited to reflect the dependence in the arrival and severity of losses generated by portfolios of insurance policies. Exploring ruin probability theoretically and numerically, under these more general dependence assumptions, is of utmost importance within the Solvency II framework of internal insolvency-risk model building.

Albrecher and Boxma (2004, 2005) have considered a collective infinite-horizon ruin model of (semi-)Markovian type where the dependence structure assures that both the consecutive claim inter-arrival times and claim sizes are respectively correlated, and there could also be a cross-correlation between them, and, as in the classical case, premiums accumulate linearly in

*Corresponding author: Faculty of Actuarial Science and Insurance, Cass Business School, City University London, 106 Bunhill Row, EC1Y 8TZ London, UK. Email: d.dimitrova@city.ac.uk

time. The model considered by Albrecher and Boxma (2005) is reasonably general and embeds the classical compound Poisson, and the Sparre-Andersen model with phase type distributed claim inter-arrival times as special cases. However, as the authors note, "in concrete cases, it is sometimes not possible to evaluate the occurring expressions". It has to be noted also that these expressions relate to the infinite-time ruin case which is not particularly relevant to finite-time applications such as modelling insurance solvency.

Under the classical constant premium rate assumption, Albrecher and Teugels (2006) consider a random walk model in which the waiting time for a claim and the claim size are dependent. Asymptotic exponential estimates for both finite and infinite time ruin probabilities are then obtained for light-tail claims, using Laplace transform. Boudreault et al. (2006) also assume that the current claim (amount) is dependent on the inter-occurrence time preceding it, and more precisely that the corresponding conditional density is defined as a mixture of two arbitrary densities with weights defined by exponentials whose powers are proportional to the preceding inter-occurrence time. In Cossette et al. (2008) the dependence between the claim amount and its corresponding inter-arrival time is modelled by a generalized Farlie-Gumbel-Morgenstern copula. The Laplace transform of the Gerber-Shiu discounted penalty function is derived, and for exponential claims, an explicit formula for the Laplace transform of the ruin time is provided. In a recent paper Sendova and Zitikis (2012) consider dependence in a risk model under the assumption that the claims arrive according to an order statistics process.

A collective finite-horizon ruin probability model with Poisson claim arrivals, dependent discrete claim amounts having any joint distribution but independent of the claim arrival times, and aggregate premium income represented by any non-decreasing positive, real valued function, has been considered by Ignatov and Kaishev (2000). They give an explicit finite-horizon ruin probability formula in terms of infinite sums of determinants which are shown by the authors to admit representation as classical Appell polynomials. Some useful properties of Appell polynomials, including a recurrence formula are given in the Appendix of that paper. An improved explicit and exact version of the ruin probability formula of Ignatov and Kaishev (2000), involving finite summation, is given in Ignatov et al. (2001). In Ignatov and Kaishev (2004), the same ruin model is considered but assuming the claim amounts have arbitrary continuous (possibly dependent) joint distribution. The finite-time ruin probability formula in that case is obtained explicitly in terms of classical Appell polynomials.

Our goal in this paper is two-fold. First, we summarize the explicit ruin probability formulas which appear in the papers by Ignatov and Kaishev (2000, 2004) and Ignatov et al. (2001), deduce new alternative expressions, and establish some enlightening connections between these formulas. The latter allow for a unified treatment and a fair comparison of their numerical efficiency. Thus, we also study the numerical properties of these formulas and propose an algorithm for their efficient evaluation with a preliminary prescribed accuracy. Based on a series of examples, we demonstrate that these formulas are useful not only theoretically but also for computing ruin probabilities in various risk models with dependence. The latter is important in practical applications. For example, as recently pointed out by Das and Kratz (2012), the need to evaluate the Ignatov-Kaishev ruin probability formulas naturally arises in the context of designing early warning systems against ruin of insurance companies. This need also arises in the context of reserving and risk capital allocation in particular, for operational risk, see Kaishev et al. (2008).

This paper is organized as follows. In section 2, we introduce our main model and give the formulas obtained by Ignatov et al. (2001) and Ignatov and Kaishev (2004) for both discrete and continuous claim amounts, and also demonstrate the interconnection between these formulas. The latter incorporates classical Appell polynomials and thus, section 3 introduces various recurrence expressions for computing classical Appell polynomials. Section 4 provides a method of computing survival probability with a prescribed accuracy and a simulation method employing order statistics proposed by Dimitrova and Kaishev (2013) is introduced in section 5. In section 6, we study the numerical properties of all theoretical results and provide several numerical examples for both discrete and continuous, dependent and independent claim severities. Section 7 concludes the paper.

2 On non-ruin probability formulas and relations between them

Let us first recall the model which we will be concerned with, which has first been considered in Ignatov and Kaishev (2000, 2004) and Ignatov et al. (2001). Let the random variables $W_1, W_2, \ldots$ denote claim severities, and let $Y_1, Y_2, \ldots$ denote their partial sums, i.e. $Y_1 = W_1, Y_2 = W_1 + W_2, \ldots$. If claim severities $W_1, W_2, \ldots, W_k$ are considered continuous random variables, then $\psi(w_1, \ldots, w_k)$ will denote their joint density and $f(y_1, \ldots, y_k)$ will de-

note the joint density of $Y_1, Y_2, \ldots, Y_k$. Clearly, $\psi(w_1, \ldots, w_k) = f(w_1, w_1+w_2, \ldots, w_1+\cdots+w_k)$ and $f(y_1, \ldots, y_k) = \psi(y_1, y_2 - y_1, \ldots, y_k - y_{k-1})$. In the case of discrete claim severities $W_1, W_2, \ldots, W_k$, their joint probability mass function $P(W_1 = w_1, \ldots, W_k = w_k)$ is denoted by $p(w_1, \ldots, w_k)$.

Let $\tau_1, \tau_2, \ldots$ denote the claim inter-arrival times assumed to be independent, identically distributed random variables, following an exponential distribution with mean $1/\lambda$, i.e. $\tau_i \sim Exp(\lambda),\ i = 1, 2, \ldots$. Thus, the number of claims up to time t is modelled by the Poisson process $N_t = \max\{i : \tau_1 + \cdots + \tau_i \leq t\},\ t > 0$. We denote by $T_1, T_2, \ldots$ the arrival times of consecutive claims, i.e. $T_i = \tau_1 + \ldots + \tau_i,\ i = 1, 2, \ldots$. Let $h(t)$ denote the premium income function of an insurance company, which is assumed a non-negative and non-decreasing real valued function defined on $\mathbb{R}_+$. It is worth noting that the condition $\lim_{t\to\infty} h(t) = +\infty$ is not necessarily required since we are interested in finite-time ruin probability. Let us also note that the function $h(t)$ does not need to be necessarily continuous. If it is discontinuous, we define $h^{-1}(y) = \inf\{z : h(z) \geq y\}$. The insurance company's surplus process is expressed as $R_t = h(t) - S_t$, where $S_t = Y_{N_t}$ is the aggregate claim amount process, and the instant of ruin T is defined as

$$T := \inf\{t : t > 0,\ R_t < 0\}$$

or $T = \infty$ if $R_t >= 0$ for all t. Under this reasonably general risk model, an exact formula for the probability of non-ruin within a finite time interval $[0, x]$, $P(T > x)$, assuming discrete claim severities, has been given in Ignatov et al. (2001), based on the formula derived in Ignatov and Kaishev (2000). The former can easily be expressed as

$$\begin{aligned} P(T > x) \;=\; & e^{-\lambda x} \sum_{k=1}^{n+1} \Bigg(\sum_{w_1=1}^{n-(k-2)} \sum_{w_2=1}^{n-(k-3)-w_1} \cdots \sum_{w_{k-1}=1}^{n-w_1-\cdots-w_{k-2}} \sum_{w_k=n+1-w_1-\cdots-w_{k-1}}^{\infty} \\ & p(w_1, \ldots, w_k) \times \sum_{j=0}^{k-1} (-1)^j b_j(\nu_1, \ldots, \nu_j) \lambda^j \sum_{m=0}^{k-j-1} \frac{(\lambda x)^m}{m!} \Bigg). \end{aligned} \tag{1}$$

where n is the integer part of $h(x)$, i.e. $n = \lfloor h(x) \rfloor$, $\nu_k = h^{-1}(w_1 + \ldots + w_k)$, and $b_j(\nu_1, \ldots, \nu_j)$ is defined recurrently as

$$b_j(\nu_1, \ldots, \nu_j) = (-1)^{j+1} \frac{\nu_j^j}{j!} + (-1)^{j+2} \frac{\nu_j^{j-1}}{(j-1)!} b_1(\nu_1) + \cdots + (-1)^{j+j} \frac{\nu_j^1}{1!} b_{j-1}(\nu_1, \ldots, \nu_{j-1}) \tag{2}$$

with $b_0 \equiv 1,\ b_1(\nu_1) = \nu_1$. In Ignatov and Kaishev (2000), b_j have been explicitly expressed as certain determinants and have been shown to admit representation as classical Appell polynomials (see the Appendix therein). It is easy to show (see Kaishev and Dimitrova 2006) that (1) generalizes to the case of continuous severities as

$$\begin{aligned} P(T>x) &= e^{-\lambda x}\sum_{k=1}^{\infty}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{k-2}}\int_{h(x)-w_1-\cdots-w_{k-1}}^{\infty}\psi(w_1,\ldots,w_k) \\ &\quad\times\Big(\sum_{j=0}^{k-1}(-1)^j b_j(\nu_1,\ldots,\nu_j)\lambda^j\sum_{m=0}^{k-j-1}\frac{(\lambda x)^m}{m!}\Big)dw_k\ldots dw_1. \end{aligned} \tag{3}$$

In Ignatov and Kaishev (2004), the following explicit formula for the probability of non-ruin, $P(T>x)$, with continuous claim severities has been derived

$$\begin{aligned} P(T>x) &= e^{-\lambda x}\Big(1+\sum_{k=1}^{\infty}\lambda^k\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{k-1}} \\ &\quad A_k(x;\nu_1,\ldots,\nu_k)\times\psi(w_1,\ldots,w_k)dw_k\ldots dw_1\Big), \end{aligned} \tag{4}$$

where $A_k(x;\nu_1,\ldots,\nu_k)$ for $k=1,2,\ldots$ are the classical Appell polynomials, $A_k(x)$, of degree k with a coefficient in front of x^k equal to $1/k!$. The classical Appell polynomials were first introduced by Appell (1880) and are uniquely defined by

$$\begin{aligned} A_0(x) &= 1, \\ A_k'(x;\nu_1,\ldots,\nu_k) &= A_{k-1}(x;\nu_1,\ldots,\nu_{k-1}), \\ A_k(\nu_k;\nu_1,\ldots,\nu_k) &= 0,\ k=1,2,\ldots, \end{aligned} \tag{5}$$

where $\nu_1 \le \ldots \le \nu_k,\ \nu_i \in \mathbb{R}$. For further background information and properties of Appell polynomials we refer the reader to e.g. Vein and Dale (1999), and their connection with the Abel–Goncharov polynomials is reviewed in e.g. Lefèvre (2007).

It can directly be seen that formula (4) is also valid for discrete claim severities in which

case it takes the form:

$$\begin{aligned} P(T>x) &= e^{-\lambda x}\Big(1+\sum_{k=1}^{n}\lambda^k \sum_{\substack{w_1\geq 1,\ldots,w_k\geq 1\\ w_1+\cdots+w_k\leq n}} A_k(x;\nu_1,\ldots,\nu_k)\times p(w_1,\ldots,w_k)\Big) \\ &\equiv e^{-\lambda x}\Big(1+\sum_{k=1}^{n}\lambda^k \sum_{w_1=1}^{n-(k-1)}\ \sum_{w_2=1}^{n-(k-2)-w_1}\cdots\ \sum_{w_k=1}^{n-w_1-\cdots-w_{k-1}} \\ &\quad A_k(x;\nu_1,\ldots,\nu_k)\times p(w_1,\ldots,w_k)\Big). \end{aligned} \tag{6}$$

In the context of ruin theory classical Appell polynomials first appear in Ignatov and Kaishev (2000). It has been shown by Ignatov and Kaishev (2000, 2004) that the occurrence of classical Appell polynomials, $A_k(x)$, in the ruin formulas (1) (through the determinants $b_j(.)$) and (4) is related to the fact that, given $N_x = k$, the random vector of claim arrival times $T_1,\ldots,T_k$ coincides in distribution with the order statistics of k independent uniformly distributed on $[0,x]$ random variables. This has later been noted in the review paper by Lefèvre and Loisel (2009) and recently, the model has more generally been named *an ordered risk model* by Picard and Lefèvre (2011). A different class of polynomials, called generalized Appell polynomials, has been used by Picard and Lefèvre (1997) to express ruin probability in the special case of i.i.d. discrete claim amounts. However, it should be noted that these polynomials do not yield the classical Appell polynomials except in the degenerate case of claim amounts equal to unity with probability 1.

Next, our purpose is to show that it is possible to derive formula (1) from (6) and vice versa, and formula (3) from (4) and vice versa. Establishing these connections has theoretical importance and it demonstrates that these formulas are very closely related through the expression of Appell polynomials in terms of the determinants b_j. However, as we will demonstrate in section 6, due to their different structures, these formulas have different numerical performance. In order to show that the formulas introduced above are inter-related and it is possible to derive one from the other, we need the following lemma.

Lemma 2.1 *The following equality holds where $b_j(.)$ and $A_i(.)$ are defined by (2) and (5) correspondingly.*

$$\sum_{j=0}^{k-1}(-1)^j b_j(\nu_1,\ldots,\nu_j)\lambda^j \sum_{m=0}^{k-j-1}\frac{(\lambda x)^m}{m!} = \sum_{i=0}^{k-1}\lambda^i A_i(x;\nu_1,\ldots,\nu_i) \tag{7}$$

Proof: For the LHS of (7), we have

$$\begin{aligned} & \sum_{j=0}^{k-1}(-1)^j b_j(\nu_1,\ldots,\nu_j)\lambda^j \sum_{m=0}^{k-j-1}\frac{(\lambda x)^m}{m!} \\ = & \sum_{j=0}^{k-1}(-1)^j b_j(\nu_1,\ldots,\nu_j) \sum_{m=0}^{k-j-1}\frac{x^m}{m!}\lambda^{m+j} \end{aligned}$$

Now redefining the inner summation index as $m = i - j$ gives

$$\begin{aligned} = & \sum_{j=0}^{k-1}(-1)^j b_j(\nu_1,\ldots,\nu_j) \sum_{i=j}^{k-1}\frac{x^{i-j}}{(i-j)!}\lambda^i \\ = & \sum_{j=0}^{k-1}\sum_{i=j}^{k-1}\lambda^i(-1)^j b_j(\nu_1,\ldots,\nu_j)\frac{x^{i-j}}{(i-j)!}, \end{aligned}$$

and changing the order of the sums leads to

$$= \sum_{i=0}^{k-1}\lambda^i \sum_{j=0}^{i}(-1)^j b_j(\nu_1,\ldots,\nu_j)\frac{x^{i-j}}{(i-j)!}.$$

Recalling eq. (A1) from Lemma 1 in Ignatov and Kaishev (2000), we note that

$$\sum_{j=0}^{i}(-1)^j b_j(\nu_1,\ldots,\nu_j)\frac{x^{i-j}}{(i-j)!} = A_i(x;\nu_1,\ldots,\nu_i)$$

which completes the proof. □

Thus, applying Lemma 2.1, formula (3) can be re-written as

$$\begin{aligned} P(T>x) = & e^{-\lambda x}\sum_{k=1}^{\infty}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{k-2}}\int_{h(x)-w_1-\cdots-w_{k-1}}^{\infty}\psi(w_1,\ldots,w_k) \\ & \times\sum_{i=0}^{k-1}\lambda^i A_i(x;\nu_1,\ldots,\nu_i)dw_k\ldots dw_1, \end{aligned} \tag{8}$$

and analogously, formula (1) can also be expressed as

$$P(T>x) = e^{-\lambda x}\sum_{k=1}^{n+1}\Big(\sum_{w_1=1}^{n-(k-2)}\sum_{w_2=1}^{n-(k-3)-w_1}\cdots\sum_{w_{k-1}=1}^{n-w_1-\cdots-w_{k-2}}\sum_{w_k=n+1-w_1-\cdots-w_{k-1}}^{\infty} p(w_1,\ldots,w_k)\times\sum_{i=0}^{k-1}\lambda^i A_i(x;\nu_1,\ldots,\nu_i)\Big). \quad (9)$$

Now we are in a position to show that formula (3) and (4) are connected.

Proposition 2.2 *Formula (3) follows from formula (4), and vice versa.*

Proof: As noted above, using Lemma 2.1 formula (3) can be alternatively expressed as (8). Substituting in (8) $k=s+1$ and changing the position of the sum with respect to i gives

$$P(T>x) = e^{-\lambda x}\sum_{s=0}^{\infty}\sum_{i=0}^{s}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{s-1}}\int_{h(x)-w_1-\cdots-w_s}^{\infty}\psi(w_1,\ldots,w_{s+1}) \times\lambda^i A_i(x;\nu_1,\ldots,\nu_i)dw_{s+1}\ldots dw_1$$

Permuting the two sums in the last expression, we obtain

$$\begin{aligned}P(T>x) &= e^{-\lambda x}\sum_{i=0}^{\infty}\sum_{s=i}^{\infty}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{s-1}}\int_{h(x)-w_1-\cdots-w_s}^{\infty}\psi(w_1,\ldots,w_{s+1}) \times\lambda^i A_i(x;\nu_1,\ldots,\nu_i)dw_{s+1}\ldots dw_1\\ &= e^{-\lambda x}\sum_{i=0}^{\infty}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{i-1}}\lambda^i A_i(x;\nu_1,\ldots,\nu_i)\\ &\times\Big(\int_{h(x)-w_1-\cdots-w_i}^{\infty}\psi(w_1,\ldots,w_{i+1})dw_{i+1}\\ &+\int_0^{h(x)-w_1-\cdots-w_i}\int_{h(x)-w_1-\cdots-w_{i+1}}^{\infty}\psi(w_1,\ldots,w_{i+2})dw_{i+2}dw_{i+1}\\ &+\int_0^{h(x)-w_1-\cdots-w_i}\int_0^{h(x)-w_1-\cdots-w_{i+1}}\int_{h(x)-w_1-\cdots-w_{i+2}}^{\infty}\psi(w_1,\ldots,w_{i+3})dw_{i+3}dw_{i+2}dw_{i+1}\\ &+\cdots\Big)dw_i\ldots dw_1 \quad (10)\end{aligned}$$

Noting that the sum in the brackets is identically equal to $\psi(w_1,\ldots,w_i)$, we have

$$\begin{aligned}P(T>x) &= e^{-\lambda x}\sum_{i=0}^{\infty}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{i-1}}\lambda^i A_i(x;\nu_1,\ldots,\nu_i)\times\psi(w_1,\ldots,w_i)dw_i\ldots dw_1\\ &= e^{-\lambda x}\Big(1+\sum_{i=1}^{\infty}\int_0^{h(x)}\int_0^{h(x)-w_1}\cdots\int_0^{h(x)-w_1-\cdots-w_{i-1}}\lambda^i A_i(x;\nu_1,\ldots,\nu_i)\times\psi(w_1,\ldots,w_i)dw_i\ldots dw\end{aligned}$$

which completes the proof. □

Similarly, we have the following proposition for the formulas with discrete claim severities.

Proposition 2.3 *Formula (1) follows from formula (6), and vice versa.*

Proof: The proof of Proposition 2.3 follows the same reasoning as the proof of Proposition 2.2, with integrals appropriately replaced by sums, and therefore it is omitted. □

In summary, we have shown that formulas (3) and (4), and formulas (1) and (6) are related and can be derived from one to another, and also that they can all be expressed in terms of classical Appell polynomials (recall that (3) is re-expressed as (8) and (1) is re-expressed as (9)), which provides for a unified treatment and direct comparisons. Thus, (8) and (9) (instead of (3) and (1)) are considered in the analysis that follows and in section 6 in particular. Although both formulas (9) and (6) (respectively (8) and (4)), represent the finite-time non-ruin probability $P(T > x)$, they differ in the way terms are accumulated in the (infinite) sum with respect to k as revealed in the proof of Proposition 2.2, particularly by (10). The main difference in the structure of formulas (9) and (6) (respectively formulas (8) and (4)) is that in formula (9) summation is over all non-ruin trajectories for which $\{Y_{k-1} \leq \lfloor h(x) \rfloor, Y_k > \lfloor h(x) \rfloor\}$, $k = 1, 2, \ldots$, whereas in formula (6) the summation index k can directly be interpreted as the number of claims up to time x and summation is over all non-ruin trajectories for which $\{N(x) = k, Y_k \leq \lfloor h(x) \rfloor\}$, $k = 1, 2, \ldots$.

In Lefèvre and Loisel (2009), the authors give the following formula for $P(T > x)$ under the same risk model for discrete claim severities (see expressions (4.10) and (4.17) therein)

$$P(T > x) = e^{-\lambda x} \sum_{i=0}^{\lfloor h(x) \rfloor} \sum_{k=0}^{i} \lambda^k E\big[I(Y_k = i) A_k(x; h^{-1}(Y_1), \ldots, h^{-1}(Y_k))\big], \tag{11}$$

where $A_k(x; h^{-1}(Y_1), \ldots, h^{-1}(Y_k))$ are the classical Appell polynomials and $Y_k = W_1 + \ldots + W_k$, $k = 0, 1, \ldots$, are the partial sums of discrete claim amounts. The authors discuss the similarity between (11) and formula (9), obtained by Ignatov and Kaishev (2000) and Ignatov et al. (2001), and note that there are some differences in the conditional events used in the derivation of the

two formulas. However, by permuting the two sums, formula (11) can be directly rewritten as:

$$\begin{aligned} P(T>x) &= e^{-\lambda x}\sum_{k=0}^{\lfloor h(x)\rfloor}\lambda^k E\big[I(Y_k\le\lfloor h(x)\rfloor)A_k(x;h^{-1}(Y_1),\ldots,h^{-1}(Y_k))\big] \\ &= e^{-\lambda x}\sum_{k=0}^{n}\lambda^k\sum_{y_1=1}^{n-(k-1)}\sum_{y_2=y_1+1}^{n-(k-2)}\cdots\sum_{y_k=y_{k-1}+1}^{n}A_k(x;\nu_1,\ldots,\nu_k)\times p(y_1,y_2-y_1,\ldots,y_k-y_{k-1}) \\ &= e^{-\lambda x}\sum_{k=0}^{n}\lambda^k\sum_{w_1=1}^{n-(k-1)}\sum_{w_2=1}^{n-(k-2)-w_1}\cdots\sum_{w_k=1}^{n-w_1-\cdots-w_{k-1}}A_k(x;\nu_1,\ldots,\nu_k)\times p(w_1,\ldots,w_k) \end{aligned}$$

The above shows that formula (11), given by Lefèvre and Loisel (2009), coincides with (6), which is a special case (discrete version) of formula (4) derived in Ignatov and Kaishev (2004).

We are further interested in the numerical properties of formulas (4) and (6), (8) and (9). Although they can be derived from one another, as we will demonstrate in section 6, they have different computational performance due to their different structures. However, their numerical efficiency crucially depends on: 1) how efficiently the Appell polynomials are computed; 2) how the (infinite) sum with respect to the number of claims k is truncated so that the resulting error is less than a pre-specified accuracy level; and 3) how efficiently the multiple sums/integrals are computed. These three aspects, affecting the numerical performance of the explicit finite-time ruin probability formulas obtained under the dependent risk model setting considered here, will be addressed in sections 3, 4 and 5 respectively.

3 On Appell polynomials

As noted, the classical Appell polynomials naturally arise in the ruin probability formulas presented in section 2. Therefore, in order to evaluate these efficiently, it is necessary to provide efficient means of computing Appell polynomials. In this section, we summarize six recurrence expressions of Appell polynomials and comment on their properties. For instance, we are interested in whether these allow for a recursive implementation only (cf. (15), (16) and (17)), or for both iterative and recursive implementations (cf. (12), (13) and (14)) in which case the flexibility in using CPU time and memory is greater. Such flexibility is preferable in ruin probability calculations, as illustrated in section 6.1. Here, we also argue that expression (14) is less computationally demanding compared to (12) and (13), and so, in the numerical study

in section 6.1 only (14),(15), (16) and (17) are considered.

We start with the recurrence expression given by Lemma 1 of Ignatov and Kaishev (2000) as

$$A_k(x;\nu_1,\ldots,\nu_k) \;=\; \sum_{i=0}^{k}\frac{x^{k-i}}{(k-i)!}A_i(0;\nu_1,\ldots,\nu_i), \tag{12}$$

where $k \geq 0$, $A_0(x)=1$, and $A_k(0;\nu_1,\ldots,\nu_k) = -\sum_{j=0}^{k-1}\frac{\nu_k^{k-j}}{(k-j)!}A_j(0;\nu_1,\ldots,\nu_j)$, $k \geq 1$.

Next, we give a similar recurrence expression for Appell polynomials

$$A_k(x;\nu_1,\ldots,\nu_k) = \sum_{i=0}^{k-1}\frac{(x^{k-i}-\nu_k^{k-i})}{(k-i)!}A_i(0;\nu_1,\ldots,\nu_i), \tag{13}$$

which easily follows from the following recurrence relation proposed by Ignatov et al. (2001) (see eq. (10) therein)

$$B_{k+1}(x;\nu_1,\ldots,\nu_k) = B_k(x;\nu_1,\ldots,\nu_{k-1}) + \sum_{i=0}^{k-1}(-1)^i\frac{b_i(\lambda\nu_1,\ldots,\lambda\nu_i)}{(k-i)!}((\lambda x)^{k-i}-(\lambda\nu_k)^{k-i}),$$

where by definition

$$B_k(x;\nu_1,\ldots,\nu_{k-1}) := \sum_{j=0}^{k-1}(-1)^j b_j(\nu_1,\ldots,\nu_j)\lambda^j\sum_{m=0}^{k-j-1}\frac{(\lambda x)^m}{m!}, k=1,2,\ldots,$$

and from Lemma 2.1, noting that

$$B_{k+1}(x;\nu_1,\ldots,\nu_k) - B_k(x;\nu_1,\ldots,\nu_{k-1}) = \lambda^k A_k(x;\nu_1,\ldots,\nu_k),$$

and that $(-1)^i b_i(\nu_1,\ldots,\nu_i)\lambda^i = A_i(0;\lambda\nu_1,\ldots,\lambda\nu_i) = \lambda^i A_i(0;\nu_1,\ldots,\nu_i)$. One can immediately see that the nature of formulas (12) and (13) is the same but in order to compute an Appell polynomial of degree $k>0$, expression (12) requires the evaluation of $k+1$ coefficients $A_j(0)$, $0 \leq j \leq k$, whereas (13) is based on k of these coefficients and thus, is preferable for computational purposes.

Further, recall that in the context of ruin theory, ν_k is defined as $\nu_k = h^{-1}(y_k)$, where y_k is the partial sums of the first k successive claims and $h(\cdot)$ is the premium income function. For instance, when the premium accumulates linearly with initial capital u and premium income

rate c, we have

$$\nu_k = h^{-1}(y_k) = \max\{0, \frac{y_k - u}{c}\}.$$

Clearly, when the initial capital u is very large, there will be a large index l such that $y_l \leq u < y_{l+1}$, which will result in $\nu_1, \ldots, \nu_l$ being equal to zero. In this case, (13) simplifies to

$$A_k(x; 0, \ldots, 0, \nu_{l+1}, \ldots, \nu_k) = \frac{(x^k - \nu_k^k)}{k!} + \sum_{i=l+1}^{k-1} \frac{(x^{k-i} - \nu_k^{k-i})}{(k-i)!} A_i(0; 0, \ldots, 0, \nu_{l+1}, \ldots, \nu_i), \quad (14)$$

where $A_k(0; 0, \ldots, 0, \nu_{l+1}, \ldots, \nu_k) = -\frac{\nu_k^k}{k!} - \sum_{j=l+1}^{k-1} \frac{\nu_k^{k-j}}{(k-j)!} A_j(0; 0, \ldots, 0, \nu_{l+1}, \ldots, \nu_j)$. Clearly, when $l \equiv 0$, (14) coincides with (13), but when the number of zero arguments is strictly positive, the former is less computationally involved.

We also introduce the following alternative recurrence expression of Appell polynomials, which utilizes their connection with the Abel–Goncharov polynomials and expresses Appell polynomials with the first l arguments $v_1 = \ldots = v_l = 0$ through Appell polynomials with no zero arguments and reduced degrees,

$$\begin{aligned} A_k(x; 0, \ldots, 0, \nu_{l+1}, \ldots, \nu_k) &= A_k(x; 0, \ldots, 0) - \sum_{j=1}^{k-l} \frac{\nu_{l+j}^{l+j}}{(l+j)!} A_{k-l-j}(x; \nu_{l+j+1}, \ldots, \nu_k) \\ &= \frac{(x^k - \nu_k^k)}{k!} - \sum_{j=1}^{k-l-1} \frac{\nu_{l+j}^{l+j}}{(l+j)!} A_{k-l-j}(x; \nu_{l+j+1}, \ldots, \nu_k). \quad (15) \end{aligned}$$

Although the above expression (15) reduces the degree of the polynomials and thus, the overall computational complexity, in contrast with (14), it has the disadvantage of being suitable only for recursive and not iterative implementation, due to its branching nature (note the dependence on the parameters v_i's). Therefore, depending on the implementation platform/software one might be preferable to the other. It should be noted that when $l = 0$, i.e. all the ν's are positive, (15) reduces to

$$\begin{aligned} A_k(x; \nu_1, \ldots, \nu_k) &= \frac{(x^k - \nu_k^k)}{k!} - \sum_{j=1}^{k-1} \frac{\nu_j^j}{j!} A_{k-j}(x; \nu_{j+1}, \ldots, \nu_k) \\ &= \frac{(x^k - \nu_k^k)}{k!} - \sum_{j=1}^{k-1} \frac{\nu_{k-j}^{k-j}}{(k-j)!} A_j(x; \nu_{k-j+1}, \ldots, \nu_k). \quad (16) \end{aligned}$$

which again is less appropriate for iterative implementation compared to e.g. (13).

Finally, another way to simplify the computation is to use the following recurrence formula which represents an Appell polynomial of degree k as a sum of lower order Appell polynomials with appropriate coefficients,

$$\begin{aligned} A_k(x;\nu_1,\ldots,\nu_k) &= \sum_{j=0}^{l-1} A_j(\nu_l;\nu_1,\ldots,\nu_j)\times\Big[\frac{(x-\nu_l)^{k-j}}{(k-j)!} - \\ &\quad \sum_{i=l+1}^{k}\frac{(\nu_i-\nu_l)^{i-j}}{(i-j)!}A_{k-i}(x-\nu_l;\nu_{i+1}-\nu_l,\ldots,\nu_k-\nu_l)\Big], \end{aligned} \tag{17}$$

where l can be any integer from 1 to k. Normally, in order to reduce the degree of the Appell polynomials to a minimum, one should select $l = \lfloor \frac{k}{2} \rfloor + 1$. Again, depending on the implementation platform, (17) might be preferable to an iterative–friendly alternative. In section 6.1, we compare the computational efficiency of these alternative expressions of Appell polynomials and also combinations of them.

4 A method for computing $P(T > x)$ with a prescribed accuracy

In this section, we introduce a method for computing the survival probability with a prespecified accuracy by truncating the number of summands in the formulas. As can be observed, formula (4), for instance, involves infinite summation which needs to be appropriately truncated from above in order to evaluate $P(T > x)$ for continuous claim severities. Furthermore, a truncation of the summation from below can also be applied since, depending on the values of the parameters, the first few summands could also be negligibly small and would not contribute much to the survival probability. Therefore, here we propose a method of truncation of the number of the summands from above and below which also leads to reduction in computation time.

It should be noted that although the evaluation of $P(T > x)$ for discrete claim severities, by using e.g. formula (6), involves finite summation and is exact, the same truncation technique can be applied to reduce the number of summands needed to be computed and thus, provide substantial efficiency gains. Therefore, in this section, we also give the discrete versions of the truncating methods.

It should also be noted that due to the different structures (and derivation) of expressions (4) and (8), and also of expressions (6) and (9), the truncation methods developed here are not (directly) applicable to (8) and (9). However, this would not be an issue since, as it is illustrated in the numerical studies in section 6, expressions (8) and (9) are less computationally appealing compared to (4) and (6), and hence, the latter are to be preferred when numerically evaluating $P(T > x)$ within the dependent, ordered risk model setting considered here.

Recall that (see e.g. Ignatov and Kaishev 2004) the probability of survival can be expressed as

$$P(T > x) = \sum_{k=0}^{\infty} P(N_x = k) P(T > x | N_x = k),$$

where N_x represents the number of claims occurring before time $x > 0$. Denote

$$P_l^m(T > x) = \sum_{k=l}^{m} P(N_x = k) P(T > x | N_x = k), \quad l = 0, 1, 2, \ldots, m = 0, 1, 2, \ldots, m \geq l.$$

Consider the remainder term in the continuous claim severities case

$$\begin{aligned} P(T > x) - P_l^m(T > x) &= \sum_{k=0}^{l-1} P(N_x = k) P(T > x | N_x = k) + \sum_{k=m+1}^{\infty} P(N_x = k) P(T > x | N_x = k) \\ &\leq \sum_{k=0}^{l-1} P(N_x = k) + \sum_{k=m+1}^{\infty} P(N_x = k) \\ &= \sum_{k=0}^{l-1} \frac{(\lambda x)^k}{k!} e^{-\lambda x} + \sum_{k=m+1}^{\infty} \frac{(\lambda x)^k}{k!} e^{-\lambda x} \end{aligned}$$

where we have used the fact that $P(T > x | N_x = k) \leq 1$.

For a given small $\epsilon > 0$, which specifies the accuracy level, denote by $l_\epsilon^* \in \mathbb{N}$ and $m_\epsilon^* \in \mathbb{N}$ the (appropriately chosen) nonnegative integers such that $\sum_{k=0}^{l_\epsilon^*-1} \frac{(\lambda x)^k}{k!} + \sum_{k=m_\epsilon^*+1}^{\infty} \frac{(\lambda x)^k}{k!} e^{-\lambda x} \leq \epsilon$. Then, we have $P(T > x) - P_{l_\epsilon^*}^{m_\epsilon^*}(T > x) \leq \epsilon$. Without loss of generality, in the numerical study in section 6 we choose to work with the largest l_ϵ^* such that $\sum_{k=0}^{l_\epsilon^*-1} \frac{(\lambda x)^k}{k!} e^{-\lambda x} \leq \frac{\epsilon}{2}$, i.e. l_ϵ^* is the $100\frac{\epsilon}{2}$-th percentile of a Poisson distribution with parameter λx, and the smallest m_ϵ^* such that $1 - \sum_{k=0}^{m_\epsilon^*} \frac{(\lambda x)^k}{k!} e^{-\lambda x} \leq \epsilon'$, where $\epsilon' := \epsilon - \sum_{k=0}^{l_\epsilon^*-1} \frac{(\lambda x)^k}{k!}$. Thus, m_ϵ^* is the $100(1-\epsilon')$-th percentile of a Poisson distribution with parameter λx. Note that it is easy to compute both l_ϵ^* and m_ϵ^* since they do not depend on the distribution of the W_i's.

Clearly, the smaller ϵ is, the smaller l_ϵ^* and the larger m_ϵ^* will be, i.e. more terms will be

needed for achieving the required accuracy of computing the survival probability. Also, l_ϵ^* and m_ϵ^* are rough bounds for the largest l_ϵ and the smallest m_ϵ such that $P(T > x) - P_{l_\epsilon}^{m_\epsilon}(T > x) \leq \epsilon$. However, these bounds may sometimes be considered as too rough since for large k, $P(T > x|N_x = k)$ may become very small, far smaller than one, because ruin will become more likely when more claims occur. One way to improve the above method is to note that $P(T > x|N_x = k) \leq P(T > x|N_x = k^*)$ and hence, consider

$$\begin{aligned} P(T > x) - P_l^m(T > x) &= \sum_{k=0}^{k^*-1} P(N_x = k)P(T > x|N_x = k) + \sum_{k=k^*}^{l-1} P(N_x = k)P(T > x|N_x = k) \\ &+ \sum_{k=m+1}^{\infty} P(N_x = k)P(T > x|N_x = k) \\ &\leq \sum_{k=0}^{k^*-1} P(N_x = k)P(T > x|N_x = k) + \sum_{k=k^*}^{l-1} P(N_x = k)P(T > x|N_x = k^*) \\ &+ \sum_{k=m+1}^{\infty} P(N_x = k)P(T > x|N_x = k^*) \\ &\leq \epsilon \end{aligned} \tag{18}$$

for a relatively small $k^* = 0, 1, 2, \ldots$. Thus, analogously one could find l_ϵ^* and m_ϵ^*, $0 \leq k^* \leq l_\epsilon^* \leq m_\epsilon^*$, such that the last inequality in (18) is fulfilled. Obviously, the larger k^* is, the closer l_ϵ^* and m_ϵ^* will be to the exact values l_ϵ and m_ϵ and the more accurate the approximation will be. However, the implementation of (18) would require computing $P(T > x|N_x = k^*)$, i.e. evaluating multiple integrals of dimension up to k^*, since the latter probability depends on the distribution of the W_i's.

As mentioned above, the same truncation technique can be applied to formula (6), in which case one could find (appropriately chosen) l_ϵ^* and m_ϵ^* such that

$$\begin{aligned} P(T > x) - P_{l_\epsilon^*}^{m_\epsilon^*}(T > x) &\leq \sum_{k=0}^{k^*-1} P(N_x = k)P(T > x|N_x = k) + \sum_{k=k^*}^{l_\epsilon^*-1} P(N_x = k)P(T > x|N_x = k^*) \\ &+ \sum_{k=m_\epsilon^*+1}^{n} P(N_x = k)P(T > x|N_x = k^*) \\ &\leq \epsilon \end{aligned} \tag{19}$$

where $n = \lfloor h(x) \rfloor$, $k^* = 0, 1, 2, \ldots$ and $k^* \leq l_\epsilon^* \leq m_\epsilon^* \leq n$. Similarly to the continuous claim severities case, when $k^* = 0$ we find the largest l_ϵ^* such that $\sum_{k=0}^{l_\epsilon^*-1} \frac{(\lambda x)^k}{k!} e^{-\lambda x} \leq \frac{\epsilon}{2}$ (i.e. l_ϵ^*

is the $100\frac{\epsilon}{2}$-th percentile of a Poisson distribution with parameter λx), and it can be shown that m_ϵ^* is the $100(1-\widetilde{\epsilon})$-th percentile of a Poisson distribution with parameter λx where $\widetilde{\epsilon} = \epsilon + \sum_{k=n+1}^{\infty} \frac{(\lambda x)^k}{k!} e^{-\lambda x} - \sum_{k=0}^{l_\epsilon^*-1} \frac{(\lambda x)^k}{k!} e^{-\lambda x}$. It should be noted though that for some choices of the parameters λ, x and $h(.)$, the additional term, $\sum_{k=n+1}^{\infty} \frac{(\lambda x)^k}{k!} e^{-\lambda x}$, could be 'large' compared to ϵ and so $m_\epsilon^* \equiv n$. But for other choices of the parameters, e.g. large initial capital u, the additional term could be far smaller than ϵ itself and therefore, the values of m_ϵ^* obtained using ϵ and using $\widetilde{\epsilon}$, i.e. in the continuous and in the discrete claim amounts case, will coincide (see section 6.3).

In general, using (19) would allow one to cut off the summation in formula (6) which, although finite, may involve the computation of terms that are negligibly small and lead to an improvement in the accuracy of digits after the decimal point which are beyond the prescribed (desired) level. In this way, evaluating Appell polynomials of a (very) high order (i.e. evaluating $A_k(.)$ in formula (6) for very large k) would be avoided and computation time would be reduced.

5 A simulation-based method for computing the high dimensional integrals/sums in $P(T > x)$ with order statistics

As mentioned above, another aspect which presents a challenge when implementing formulas (4) and (6) numerically is that they incorporate multivariate integration/summation with an increasing dimension k, which could be very computationally intensive in high dimensions. Note that the challenge in the case of discrete claim amounts is related to the fact that for a fixed dimension k and value of $n = \lfloor h(x) \rfloor$, the number of terms in formula (6) which have to be evaluated and summed up is $\binom{n}{k}$. The latter could be very large if n and/or k are large. Hence, depending on the choice of parameters, a direct implementation of (4) and (6), using a specialized software, may sometimes be highly time and memory consuming and therefore, impractical. In what follows, we introduce a simulation-based method, proposed by Dimitrova and Kaishev (2013), which employs order statistics of uniforms to compute the high dimensional integrals (or sums) in the ruin probability formulas presented here. We shall use the representation of $P(T > x)$ in terms of the partial sums $Y_1, \ldots, Y_k$.

As has been established by Dimitrova and Kaishev (2013), formula (4) for the case of

continuous claim amounts can be rewritten as

$$P(T>x) = e^{-\lambda x}\Big(1+\sum_{k=1}^{\infty}\lambda^k\,V_k^c\int_0^{h(x)}\int_{y_1}^{h(x)}\cdots\int_{y_{k-1}}^{h(x)}A_k(x;h^{-1}(y_1),\ldots,h^{-1}(y_k))\times f(y_1,\ldots,y_k) \times\frac{1}{V_k^c}\,dy_k\ldots dy_1\Big), \tag{20}$$

where

$$V_k^c=\int_0^{h(x)}\int_{y_1}^{h(x)}\cdots\int_{y_{k-1}}^{h(x)}1dy_k\ldots dy_1=\frac{h(x)^k}{k!}.$$

Similarly, in the case of discrete claim amounts formula (6) can be rewritten as

$$P(T>x) = e^{-\lambda x}\Big(1+\sum_{k=1}^{n}\lambda^k\,V_k^d\sum_{y_1=1}^{n-(k-1)}\sum_{y_2=y_1+1}^{n-(k-2)}\cdots\sum_{y_k=y_{k-1}}^{n}A_k(x;h^{-1}(y_1),\ldots,h^{-1}(y_k)) \times p(y_1,y_2-y_1,\ldots,y_k-y_{k-1})\times\frac{1}{V_k^d}\Big), \tag{21}$$

where $n=\lfloor h(x)\rfloor$ and

$$V_k^d=\sum_{y_1=1}^{n-(k-1)}\sum_{y_2=y_1+1}^{n-(k-2)}\cdots\sum_{y_k=y_{k-1}}^{n}1=\frac{n!}{k!(n-k)!}.$$

Obviously, $1/V_k^c$ (respectively, $1/V_k^d$) can be viewed as the joint probability density (respectively, mass) function of the order statistics $Y_{(1)},Y_{(2)},\ldots,Y_{(k)}$, of k independent uniformly distributed on $[0,h(x)]$ (respectively, on $\{1,2,\ldots,n\}$) random variables, see e.g. Karlin and Taylor (1981), Ch. 13 (respectively, see e.g. Arnold et al. (2008), section 3.7). Therefore, the multiple integrals (or sums) in (20) (or (21)) can be interpreted as expectations with respect to the order statistics $Y_{(1)},Y_{(2)},\ldots,Y_{(k)}$ and so, Monte-Carlo simulation approach can be used to evaluate these. Thus, one may simulate, say $N>0$, samples from the k-tuples $Y_{(1)},Y_{(2)},\ldots,Y_{(k)}$ and utilizing the method introduced in section 4, for truncating the infinite sum so that a prescribed accuracy is achieved, we arrive at

$$P(T>x)\approx e^{-\lambda x}\Big(1+\sum_{l=1}^{N}\sum_{k=\max(l_\epsilon^*,1)}^{m_\epsilon^*}\lambda^k\,V_k^c\,A_k(x;h^{-1}(y_{(1)}^l),\ldots,h^{-1}(y_{(k)}^l))\times f(y_{(1)}^l,\ldots,y_{(k)}^l)\Big)/N \tag{22}$$

or in the discrete claim amounts case at

$$P(T>x) \approx e^{-\lambda x}\Big(1+\sum_{l=1}^{N}\sum_{k=\max(l_\epsilon^*,1)}^{m_\epsilon^*}\lambda^k\times V_k^d\times A_k(x;h^{-1}(y_{(1)}^l),\ldots,h^{-1}(y_{(k)}^l)) \times p(y_{(1)}^l, y_{(2)}^l-y_{(1)}^l,\ldots,y_{(k)}^l-y_{(k-1)}^l)\Big)/N. \quad (23)$$

There are several methods proposed in the literature of simulating order statistics from a uniform distribution on an interval $[0,h(x)]$ with some being more computationally expensive than others, see e.g. Arnold et al. (2008), sections 4.7 and 4.10 therein. For the purpose of evaluating (22), we have chosen to work with the so-called descending method (cf. Arnold et al. 2008, section 4.10), since it is fast and avoids sorting, it places high importance on the large $y_{(k)}$'s, and is also quasi-Monte Carlo friendly. Also, there are several methods proposed in the literature for simulating the discrete order statistics on $\{1,2,\ldots,n\}$ in (23). We have chosen to work with the sequential random sampling Method A, as described by Vitter (1987), which avoids sorting and is an improved version of the so-called Algorithm S (cf. Knuth 1997, section 3.4.2) as it only requires the generation of k uniform variates.

In summary, in order to compute the survival probability $P(T>x)$ using formulas (4) and (6), the following procedure can be implemented.

1. For a given $\epsilon>0$, apply the method introduced in section 4 and find l_ϵ^* and m_ϵ^*, $0\le l_\epsilon^*\le m_\epsilon^*\le n$, to be used in evaluating (22) or (23). Set $l=1$.

2. For each k, such that $\max(l_\epsilon^*,1)\le k\le m_\epsilon^*$, generate k ordered uniformly distributed random variables, $y_{(1)}^l,\ldots,y_{(k)}^l$.

3. For $y_{(1)}^l,\ldots,y_{(k)}^l$, calculate the value of the l-th summand in (22) or (23), which involves computing the sum with respect to k.

4. Repeat steps 2-3 for $l=2,\ldots,N$, where N is a sufficiently large number of simulations. The non-ruin probability is then estimated by averaging over the N simulated values.

Two advantages of this algorithm over the direct Monte Carlo simulation of (non-)ruin probabilities are worth pointing out. First, we note that this algorithm is universal. In contrast to the direct MC simulation, it only requires generation of uniforms and there is no need to simulate from a particular joint distribution of the claims, which may be a formidable task especially

when the underlying claim dependence structure is complex. Second, as will be illustrated in section 6.4, for some sets of values of the parameters the above algorithm could achieve more accurate and stable results compared to the slowly convergent direct MC simulation of ruin probabilities.

6 Numerical study

This section is devoted to studying the numerical properties of the non-ruin probability formulas presented in section 2. In particular, we explore how the different recurrence expressions for Appell polynomials given in section 3 and the numerical algorithms described in sections 4 and 5 affect the efficiency of computing $P(T > x)$ using formulas (4) and (6), (8) and (9). For the purpose, we have used *Mathematica* system and a standard PC with 2.93 GHz Intel(R) Core(TM) i7 CPU and 8.00GB RAM.

This section is organized as follows. In section 6.1, we study the numerical performance of the discrete formulas (6) and (9) with respect to the alternative ways of computing Appell polynomials through the recursions summarized in section 3 and draw conclusions on the efficiency of the Appell polynomial representations. Then, utilizing the results from section 6.1, in section 6.2 we compare the different summation structures embedded in formulas (6) and (9), (4) and (8), and argue that (6) and (4) are computationally more appealing compared to (9) and (8). The numerical performance of the method of computing survival probabilities with a prescribed accuracy is investigated in section 6.3. Finally, in section 6.4, we illustrate the numerical efficiency of formulas (4) and (6) and of the simulation-based method described in section 5 which employs order statistics to compute the ruin probability.

6.1 Computing the classical Appell polynomials

This section is devoted to comparing the computational efficiency of the alternative expressions for Appell polynomials introduced in section 3. We study their numerical performance on the basis of the computational time required to compute $P(T > x)$ using formulas (6) and (9). We note that these formulas involve finite summations and are therefore exact, i.e. the accuracy is (theoretically) infinite, and so, comparing the computational time is sufficient when these are implemented directly (i.e. without any additional truncation of summands). It is also worth

noting that for the purpose of this numerical study, we have implemented all the recurrence expressions of Appell polynomials introduced in section 3 and all their possible combinations, paying attention to the order in which they combine with one another. However, here we only present the results related to expressions and combinations of them which are essential or perform better than the others. The following example with discrete i.i.d claim sizes and linear premium income function is considered in order to avoid obscuring the resulting CPU times, and hence, the comparison, by adding extra complexity.

Example 6.1 *Consecutive claim severities are assumed to follow an i.i.d. logarithmic distribution with parameter* α*, i.e.* $W \sim Log(\alpha)$ *with a generic p.m.f.* $P(W = i) = -\alpha^i/(i \ln(1-\alpha))$*, and a linear premium income function* $h(t) = u + ct$*.*

Table 1 compares the computational efficiency of the alternative recurrence expressions of Appell polynomials (14), (15), (16) and (17), and their combinations on the basis of computing $P(T > x)$ using formula (9). The comparison is done by implementing expression (14) both iteratively and recursively, whereas the rest of the expressions have only been implemented recursively since, as it has been discussed in section 3, they are not directly suitable for iterative implementation, i.e. using a looping control structure, due to their branching nature. It is worth mentioning that *Mathematica* allows for the output results of a recursion to be automatically stored in the RAM. The latter feature plays an important role in decreasing significantly the computational time when evaluating expressions (14), (15), (16) and (17) recursively.

The computation times given in Table 1 illustrate the trade-off between memory use and CPU overhead in *Mathematica*, but overall the differences are not substantial. Thus, depending on the software and the specifications of the system used to implement the survival probability formulas, different expressions for computing the Appell polynomials could be preferable. Our experience shows that generally, recursive implementations use more memory and should be avoided when computing $P(T > x)$ with large values of n because of potential stack overflow problems.

[Place Table 1 about here]

Table 2 provides the results for the same numerical comparison but on the basis of computing $P(T > x)$ using formula (6).

[Place Table 2 about here]

Comparing Tables 1 and 2, it is clear that for the specified Example 6.1 the computational times associated with formula (6) are consistently lower than that of formula (9). Our extensive trials show that this relation always holds (see also Table 3 in the next section). One can therefore argue that (6), and consequently (4), have more computationally appealing summation structure. In addition, the truncation method developed in section 4 can also be applied to (6) to further improve CPU time (cf. section 6.3) - recall that the method is not directly applicable to (8) and (9). Hence, expression (6) is computationally more appealing when evaluating survival probabilities assuming discrete claim amounts. Next, we demonstrate that similarly, in the continuous claim severities case, formula (4) is preferable to (8).

6.2 Comparing formulas (6) and (9), (4) and (8)

In this section, based on the results presented in section 6.1, we choose to work with the iterative implementation of the Appell polynomial expression (14) when comparing the survival probability formulas (6) and (9), (4) and (8) and examining the computational efficiency of their different summation structures.

First, in Figure 1, we illustrate the computational time needed to evaluate $P(T > x)$ using formulas (6) and (9) with the assumptions of Example 6.1 and parameter values as in Tables 1–2 with u and x varying. Time is measured in seconds and plotted on a log scale against a range of values for the initial capital u and the time horizon x.

[Place Figure 1 about here]

It can be seen that formula (6) is computationally more efficient than formula (9) for the whole range of values of u and x illustrated in Figure 1. More precisely, the difference in the absolute CPU times is increasing (it is less than 1 sec for $u = 10$ and $x = 5$, and increases to about 60 sec when $u = 10$ and $x = 10$) while the relative difference remains about the same, namely (6) is about 1.3 times faster than (9) for the chosen values of the parameters. Recall that both formulas are exact and can be derived from one another. So, the difference in the numerical efficiency is due to their different summation structures. Figure 1 also demonstrates that the computational time is somewhat more sensitive to the length of the time horizon x than

the size of the initial capital u which will be reconfirmed in section 6.3. The above conclusions are supported by all the results we have obtained when implementing numerically the formulas with different parameter values and assumptions for the joint distribution of the claim amounts.

Recall that both (6) and (9) can be used to compute survival probability for integer as well as non-integer initial capital u and time horizon x. It is interesting to observe that the CPU time increases stepwise. The "steps" occur when n increases, where $n = \lfloor h(x) \rfloor = \lfloor u+cx \rfloor$ is the integer part of the premium income. This is anticipated, because n represents the dimension of the outer sum in the corresponding formulas.

Next, we turn our attention to the computational efficiency of formulas (4) and (8). We compare the latter on the basis of the following example.

Example 6.2 *Consecutive claim severities are assumed to follow an i.i.d. exponential distribution with parameter α, i.e. $W \sim Exp(\alpha)$ with a generic p.d.f. $\psi(w) = \alpha e^{-\alpha w}$, and a linear premium income function $h(t) = u + c\,t$.*

As both formulas involve a sum to infinity, we are interested in finding the number of summands, m_ϵ, which should be evaluated in order to achieve an accuracy of 4 (or 6) correct digits after the decimal point. Some numerical results are presented in Table 3, where a comparison between the computational efficiency and accuracy of formulas (4) and (8), is drawn.

[Place Table 3 about here]

For all trials illustrated in Table 3, formula (4) evaluated with $m_\epsilon = 5$ gives values for $P(T > x)$ with all six digits after the decimal point correct (for comparison, these values could also be obtained using some of the known explicit expressions for this classical risk model with exponential claim amounts, see e.g. Chapter V in Asmussen and Albrecher 2010 or Garcia 2005). Furthermore, if we require 4 correct digits after the decimal point, we see that evaluating formula (4) with $m_\epsilon = 4$ is sufficient to achieve this accuracy level. However, if we are to use formula (8), m_ϵ needs to be 8 or 9 to reach the same level of accuracy, which means that the latter formula converges much slower with respect to k. Thus, more terms are required in the summation in formula (8) to achieve a desired accuracy, and this affects the computation time substantially, as can be seen from Table 3. The latter conclusion is reconfirmed by all the

results we have obtained when implementing (4) and (8) with various parameter values and assumptions about the joint distribution of the claims.

In summary, although we have proved formulas (6) and (9), (4) and (8) can be derived from one another, it is observed that their numerical performance is quite different. In the discrete case, without truncating the summation, both formulas are exact and, as observed in Figure 1, formula (6) is more computationally appealing although the difference is relatively small. In the continuous case, it appears that formula (4) is computationally more efficient than (8) as it requires fewer terms in order to achieve certain fixed level of accuracy.

6.3 Computing $P(T > x)$ with a prescribed accuracy

This section explores the numerical performance of the method of computing survival probability with a prescribed accuracy introduced in section 4. Both cases of continuous and discrete claim severities are investigated.

First, we study the numerical performance of (19) which provides a method for truncating the summation in (6) from both below and above. Thus, Table 4 illustrates the truncating points l_ϵ^* and m_ϵ^* for a range of values of λx and $n = \lfloor h(x) \rfloor$ and a chosen level of accuracy, ϵ. As can be seen, the truncation interval given by l_ϵ^* and m_ϵ^* widens with the increase in the required level of accuracy.

[Place Table 4 about here]

In Table 4, it is observed that, for fixed values of ϵ and k^*, both the lower and the upper truncating points increase, subject to a maximum of $n = \lfloor h(x) \rfloor$, with the increase in length of the time horizon x and/or the Poisson rate λ. Furthermore, it is observed that, for fixed λx, the lower truncating point remains the same with the increase in n (recall that determining l_ϵ^* from (19) with $k^* = 0$ depends only on λx) and the upper truncating point increases, subject to a maximum determined by formula (18).

As explained in section 4, formula (18) defines upper and lower truncation points in the case of continuous claim amounts, which for $k^* = 0$ solely depends on the product λx. Thus, Table 5 summarizes the truncating points l_ϵ^* and m_ϵ^*, estimated using (18), for a range of values of λx and a chosen level of accuracy, ϵ.

[Place Table 5 about here]

As explained in section 4, and also seen in the calculations related to Tables 4 and 5, it is straightforward to obtain estimates of the truncating points with $k^* = 0$, as specified by (18) and (19), because these estimates do not depend on the distribution of the claim amounts.Recall that l_ϵ^* and m_ϵ^* could be further refined if $k^* > 0$ is used so that formulas (18) and (19) take into account the distribution of the W_i's.

Tables 6 and 7 illustrate the truncating point from above, m_ϵ^*, estimated using (18) with $k^* > 0$ and level of accuracy ϵ, evaluated under the assumptions of Example 6.2 and certain chosen values of the parameters λ, α, u, c, x.

[Place Table 6 about here]

[Place Table 7 about here]

In Table 6, it is illustrated that, for a fixed level of accuracy $\epsilon = 10^{-3}$, the upper truncation point of the sum in (4) could be refined by using $k^* > 0$ in (18). Although the decrease in m_ϵ^* may not seem significant, we observe that the computation time is more than halved when the highest dimension of the integrals in (4) is reduced from 5 to 4. Trials with different parameter values show that in some cases when u and/or x are large, cutting off one more term may lead to even more considerable reduction in CPU time. Finally, it is also observed that the values of $P(T > x)$ presented in Table 6 have four correct digits when $m_\epsilon^* = 4$ and six correct digits when $m_\epsilon^* = 5$ although both of these are estimated using (18) with a required level of accuracy of $\epsilon = 10^{-3}$. Therefore, using $k^* > 0$ in (18) results in a more precise estimate of m_ϵ^*.

Table 7 demonstrates that higher level of accuracy is associated with larger number of summands, m_ϵ^*, to be evaluated in (4), and hence longer time to compute $P(T > x)$. The fact that the approximated survival probability values always have more correct digits than the required level of accuracy is evident from Table 7. The latter should be taken into account when evaluating $P(T > x)$ using (4) since the difference in computation time for e.g. $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$ could be significant.

6.4 On the simulation-based method for computing $P(T > x)$ with order statistics

In this section, we study the numerical performance of the simulation-based method for computing the high dimensional integrals/sums in (4) and (6) with order statistics, as introduced in section 5 for both discrete and continuous, dependent and independent claim severities. On the basis of various examples provided here, the numerical efficiency of this method is compared to that of the direct MC simulation of the finite-time survival probability. It is worth mentioning that for the former we first utilize the truncation method described in section 4, in order to achieve a prescribed accuracy of ϵ, and then apply a hybrid approach, where we directly compute the integrals/sums with low dimensions (e.g. up to 5/10) that are easily computable and approximate the remaining dimensions with the algorithm employing order statistics. It should be noted that this algorithm is quasi-MC friendly which in general is not the case for direct MC simulation of $P(T > x)$. Thus, one could further improve the accuracy of the final estimate by simulating wherever needed uniforms from a low discrepancy sequence. In our examples we employ the well-known Sobol sequence (a C++ code written by John Burkardt for generating Sobol' sequences up to dimension $1,111$ is downloadable from `http://people.sc.fsu.edu/~jburkardt/cpp_src/sobol/sobol.html`).

We first illustrate the numerical performance of this order statistics simulation-based method in the discrete claim amounts case and contrast it with the direct MC simulation approach, under the assumptions of Example 6.1. For each of the two simulation approaches, in order to obtain an estimate of the survival probability $P(T > x)$ and an estimate of its variance, we run $M = 100$ independent replications where each is based on N number of simulations. This is done for $N = 1 \times 10^4$, 2×10^4, 5×10^4, 1×10^5. Note that in this classical risk model with discrete i.i.d. claim sizes and for the chosen values of the parameters, the exact value of $P(T > x)$ can be evaluated using the Picard-Lefèvre formula and utilizing an extension of the Panjer recursions as shown by De Vylder (1999). Thus, knowing the exact value of the survival probability, the absolute bias of the simulated estimates and their corresponding variances, with four different sets of values for the parameters, are illustrated in Figures 2 and 3.

[Place Figure 2 about here]

[Place Figure 3 about here]

In the examples illustrated in Figures 2 and 3, the required accuracy level is $\epsilon = 10^{-4}$, and the parameter sets are chosen so that in both cases this yields the same number of summands in (23), namely $m_\epsilon^* = 15$, and thus, results are somewhat directly comparable. As expected, for both methods the absolute bias and variance of the estimate decrease as the number of simulations, N, increases but it is observed that in most cases the order statistics method achieves significantly higher accuracy already for small number of simulations N and is generally more stable. Although the same number of summands ($m_\epsilon^* = 15$) are evaluated in each of the four cases illustrated in Figure 2 (a) and (b), and Figure 3 (a) and (b), the difference in the behavior of the order statistics method comes from the fact that the distribution of the probability mass across the summands is different in all four cases and the absolute value of each summand also changes. In general, it can be said that when the absolute values of the summands are 'small' or when there is substantial right skewness in the distribution of the probability mass across the summands then the order statistics method significantly outperforms the direct simulation method, see e.g. the left panel of Figure 2 (a), where the relative error of the estimate is in the range of $0.0025\% - 0.0006\%$ and $3.58\% - 1.04\%$ respectively for the two methods, or the left panel of Figure 3 (a), where the relative error is in the range of $0.0004\% - 0.0001\%$ and $0.66\% - 0.23\%$ respectively. However, when the absolute values of the summands are 'large' or when there is substantial left skewness in the distribution of the probability mass across the summands then the accuracy and the volatility of the order statistics method are comparable to that of the direct simulation (see e.g. the left panel of Figure 3 (b), where the relative error is in the range of $0.1\% - 0.03\%$ and $0.06\% - 0.02\%$ respectively) or only slightly better than that of the direct simulation (see e.g. the left panel of Figure 2 (b), where the relative error is in the range of $0.23\% - 0.05\%$ and $0.89\% - 0.30\%$ respectively).

Next, we consider again discrete claim amounts but with dependence, as specified in Example 6.3 below. The dependence structure considered here is Clayton copula, which is suitable for modelling lower tail dependence.

Example 6.3 *We assume dependent claim severities whose distribution is modelled by a Clayton copula with parameter θ and logarithmic marginals with parameter α, i.e. $W \sim Log(\alpha)$ with a generic p.m.f. $P(W = i) = -\alpha^i/(i \ln(1-\alpha))$. The premium income function is assumed*

linear, i.e. $h(t) = u + c\,t$.

Recall that (see e.g. Chapter 5 of McNeil et al. 2005) for Clayton copula, we have

$$P(W_1 \leq w_1, \ldots, W_k \leq w_k) \equiv C^{CL}(u_1, \ldots, u_k; \theta) = \left(\sum_{s=1}^{k} u_s^{-\theta} - k + 1 \right)^{-1/\theta} = \left(\sum_{s=1}^{k} (p_s(w_s))^{-\theta} - k + 1 \right)^{-1/\theta},$$

where $P(W_s \leq w_s) \equiv p_s(w_s)$, $s = 1, \ldots, k$. Thus, to determine the joint probability mass function of the consecutive claim amounts, $p(w_1, \ldots, w_k)$, which appear in (6) (and (23)), we have (cf. Panagiotelis et al. 2012)

$$\begin{aligned} p(w_1, \ldots, w_k) &= \sum_{\epsilon_1=0}^{1} \cdots \sum_{\epsilon_k=0}^{1} (-1)^{\epsilon_1 + \cdots + \epsilon_k} P(W_1 \leq w_1 - \epsilon_1, \ldots, W_k \leq w_k - \epsilon_k) \\ &= \sum_{\epsilon_1=0}^{1} \cdots \sum_{\epsilon_k=0}^{1} (-1)^{\epsilon_1 + \cdots + \epsilon_k} \left(\sum_{s=1}^{k} (p_s(w_s - \epsilon_s))^{-\theta} - k + 1 \right)^{-1/\theta}. \end{aligned}$$

[Place Figure 4 about here]

As can be seen in Figure 4, the relative behavior of the discrete order statistics method and direct simulation method is very similar to that observed in Figure 3 where i.i.d. logarithmic claim amounts are assumed. Hence, the relative performance of the order statistic method is not so much influenced by introducing a dependence structure. Our experience shows that the relative behavior is also not significantly affected when a non-linear premium income function is considered (as done in Example 6.5). Note that on this example one could only directly compare the simulated estimates of $P(T > x)$ instead of their absolute bias since the exact value of $P(T > x)$ is unknown and is difficult to evaluate with an accuracy of 4 correct digits using formula (6) as $m_\epsilon^* = 15$.

Next, we turn our attention to the case of continuous claim severities. For the sake of brevity, we do not give details of the numerical performance of the order statistics simulation-based method with the assumptions of Example 6.2 (i.i.d. exponentially distributed claims) as the results are somewhat similar (or better) to those of Example 6.1. Instead, we consider continuous claim amounts with dependence, as specified in Example 6.4 below. We note that the dependence structure of the rotated Clayton copula is suitable for modelling upper tail

dependence, e.g. as in catastrophic events (cf. Chapter 5 and 8 of McNeil et al. 2005), and the assumed Pareto distribution for the marginals is heavy-tailed.

Example 6.4 *We assume dependent claim severities whose distribution is modelled by a rotated Clayton copula with parameter θ and Pareto marginals with parameters λ_p and α_p, i.e. $W \sim Pareto(\lambda_p, \alpha_p)$ with a generic p.d.f.*

$$\psi(w) = \frac{\alpha_p \lambda_p^{\alpha_p}}{(\lambda_p + w)^{\alpha_p+1}}.$$

The premium income function is assumed linear, $h(t) = u + c\,t$.

Given the assumptions of Example 6.4, in order to determine the joint density function of the consecutive claims amounts, $\psi(w_1, \ldots, w_k)$, which appears in (4) (and (22)), we recall some well-known results on copulas. For an n-dimensional joint distribution function G with marginals $G_1, \ldots, G_n$, if there is a copula C such that $G(x_1, \ldots, x_n) = C(G_1(x_1), \ldots, G_n(x_n))$ holds for all $(x_1, \ldots, x_n)$, then the joint density function can be expressed as

$$\begin{aligned} g(x_1, \ldots, x_n) &= \frac{\partial G(x_1, \ldots, x_n)}{\partial x_1 \ldots \partial x_n} \\ &= \frac{\partial C(u_1, \ldots, u_n)}{\partial u_1 \ldots \partial u_n} \times \prod_{i=1}^{n} \frac{\partial G_i(x_i)}{\partial x_i} \\ &= c(u_1, \ldots, u_n) \times \prod_{i=1}^{n} g_i(x_i), \end{aligned}$$

where g_i is the density function of G_i, $u_i = G_i(x_i)$, for $i = 1, \ldots, n$, and $c(.)$ is the density function of copula $C(.)$. Thus, formula (4) can be re-written as

$$\begin{aligned} P(T > x) = e^{-\lambda x}\Big(1 + \sum_{k=1}^{\infty} \lambda^k \int_0^{h(x)} \int_0^{h(x)-w_1} \cdots \int_0^{h(x)-w_1-\cdots-w_{k-1}} \\ A_k(x; \nu_1, \ldots, \nu_k) \times c(G_1(w_1), \ldots, G_k(w_k)) \times \prod_{i=1}^{k} g_i(w_i) dw_k \ldots dw_1\Big), \end{aligned}$$

where g_i, $i = 1, \ldots, k$, are the density functions of the marginals of the joint distribution of consecutive claim severities.

Recall that in contrast to Clayton copula, which models lower tail dependence, rotated

Clayton copula models upper tail dependence. Based on the density of Clayton copula,

$$c^{\mathrm{Cl}}(u_1,\ldots,u_k;\theta)=\theta^k\frac{\Gamma(1/\theta+k)}{\Gamma(1/\theta)}\Big(\prod_{i=1}^{k}u_i^{-\theta-1}\Big)\Big(\sum_{i=1}^{k}u_i^{-\theta}-k+1\Big)^{-1/\theta-k},$$

the density of rotated Clayton copula can be obtained as $c^{\mathrm{RCl}}(u_1,\ldots,u_k;\theta)=c^{\mathrm{Cl}}(1-u_1,\ldots,1-u_k;\theta)$ for $\theta\in(0,\infty)$, where $\theta=0$ corresponds to independence, which is the same as for Clayton copula.

Again, we compare the order statistics method with direct MC simulation of the survival probability. The results are summarized in Figure 5.

[Place Figure 5 about here]

In the left panels of Figure 5, the estimated survival probabilities are plotted against the number of simulations obtained with the two alternative ways of simulation. In the left panel of Figure 5 (a), we also plot the exact value of the survival probability for reference, computed using (4) with $m_\epsilon^*=9$. However, in the left panel of Figure 5 (b) the exact value of $P(T>x)$ is not indicated, since it is unknown and is difficult to evaluate with an accuracy of 4 correct digits using formula (4), note that $m_\epsilon^*=15$. Therefore, on this example one could only directly compare the simulated estimates instead of their absolute bias. However, from the fluctuations of the two lines in Figures 5 (a) and (b), it can be asserted that the simulated values with both methods converge to the true value which should be within the range of the undulation of the red line, corresponding to the order statistics method. Furthermore, as can be seen in the right panels of Figure 5, the variance of the simulated values becomes smaller as the number of simulations increases, with the estimates of the order statistics method being somewhat less volatile for the set of parameters illustrated in (a) and (b).

Lastly, we consider again continuous claim amounts with dependence but with a non-linear premium income function, $h(t)$, as specified in Example 6.5.

Example 6.5 *Same as Example 6.4 but with a piece-wise linear premium income function,* $h(t)=u+0.2\sum_{i=1}^{5}\mathbb{I}_{\{i\le t\}}+c\,t$, *i.e. with a jump of size* 0.2 *at each integer* $t=1,2,3,4,5$.

[Place Figure 6 about here]

As can be seen in Figure 6, the relative behavior of the order statistics method and direct simulation method is similar to that observed in Figure 5 (b) and hence, is not significantly affected by the change in the premium income function.

7 Conclusion

We have shown that the survival probability formulas derived by Ignatov and Kaishev (2000), Ignatov et al. (2001), and Ignatov and Kaishev (2004) can be derived from one another both in their discrete and continuous versions and thus, can all be expressed in terms of the classical Appell polynomials. Various recurrence expressions for computing these polynomials have been presented, and their numerical properties have been investigated. Furthermore, the numerical efficiency of formulas (9) and (8), and (6) and (4) have been investigated and conclusions have been drawn about the differences in their summation structure. Thus, it has been demonstrated that formulas (6) and (4) are significantly more computationally efficient that (9) and (8).

A method of computing the survival probability with a prescribed accuracy using (4) has been introduced, which is also applicable for the case of discrete claim amounts whereby the number of summands in the already finite summation in (6) can be further reduced. We also studied the order statistics simulation-based method, proposed by Dimitrova and Kaishev (2013), for evaluating the survival probability formulas and provided several numerical examples to demonstrate its performance in dealing with different cases of discrete or continuous, dependent or independent claim severities. In our numerical study, we have not taken advantage of the possibilities for parallelizing the simulations and thus achieving substantial reduction in computation time but it is clear that this is easily achievable. Finally, further improvements of the accuracy of the proposed method could also be achieved borrowing ideas from the importance sampling simulation area of research, which is the subject of ongoing work.

Acknowledgements

The authors would like to thank Zvetan Ignatov for many constructive discussions which helped improve the paper. The third author also acknowledges the financial support received by Cass Business School, City University London during the course of his PhD.

References

Albrecher, H., J. L. Teugels. 2006. Exponential behavior in the presence of dependence in risk theory. *Journal of Applied Probability, 43*(1), 257–273.

Albrecher, H., O. Boxma. 2004. A ruin model with dependence between claim sizes and claim intervals. *Insurance: Mathematics and Economics, 35*(2), 245–254.

Albrecher, H., O. Boxma. 2005. On the discounted penalty function in a Markov-dependent risk model. *Insurance: Mathematics and Economics, 37*(3), 650–672.

Appell, P. 1880. Sur une classe de polynômes. *Annales scientifiques de l'cole Normale Suprieure 9*, 119–144.

Arnold, B. C., N. Balakrishnan, H. N. Nagaraja. 2008. *A First Course in Order Statistics.* Classics in Applied Mathematics 54, SIAM-Society for Industrial and Applied Mathematics, Philadelphia.

Asmussen, S., H. Albrecher. 2010. *Ruin probabilities.* Second edition. World Scientific, New Jersey.

Boudreault, M., H. Cossette, D. Landriault, E. Marceau. 2006. On a risk model with dependence between interclaim arrivals and claim sizes. *Scandinavian Actuarial Journal, 2006*(5), 265–285.

Cossette, H., E. Marceau, F. Marri. 2008. On the compound Poisson risk model with dependence based on a generalized Farlie-GumbelMorgenstern copula. *Insurance: Mathematics and Economics, 43*(3), 444–455.

Das, S., M. Kratz. 2012. Alarm system for insurance companies: A strategy for capital allocation, *Insurance: Mathematics and Economics, 51*(1), 53–65.

Dimitrova, D.S., V.K. Kaishev. 2013. On the infinite-time ruin and the distribution of the time to ruin. *Working paper.* Paper presented at the 10^{th} IME conference, Leuven, Belgium, July 18–20 2006.

De Vylder, F. E. 1999. Numerical Finite-Time Ruin Probabilities by the Picard-Lefevre Formula. *Scandinavian Actuarial Journal, 1999*(2), 97–105.

Garcia, J.M.A. 2005. Explicit solutions for survival probabilities in the classical risk model. *ASTIN Bulletin, 35*(1), 113–130.

Ignatov, Z.G., V.K. Kaishev. 2000. Two-sided Bounds for the Finite-time Probability of Ruin. *Scandinavian Actuarial Journal, 2000*(1), 46–62.

Ignatov, Z.G., V.K. Kaishev. 2004. A Finite-Time Ruin Probability Formula For Continuous Claim Severites. *Journal of Applied Probability, 41*(2), 570–578.

Ignatov, Z.G., V.K. Kaishev, R.S. Krachunov. 2001. An Improved Finite-time Ruin Probability Formula and its "Mathematica" Implementation. *Insurance: Mathematics and Economics, 29*(3), 375–386.

Kaishev, V.K., D.S. Dimitrova. 2006. Excess of Loss Reinsurance Under Joint Survival Optimality. *Insurance: Mathematics and Economics, 39*(3), 376–389.

Kaishev, V.K., D.S. Dimitrova, Z.G. Ignatov. 2008. Operational Risk and Insurance: A Ruin-probabilistic Reserving Approach. *Journal of Operational Risk, 3*(3), 39–60.

Karlin, S., H.M. Taylor 1981. *A Second Course in Stochastic Processes.* Academic Press, New York.

Knuth, D. E. 1997. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd edition, Addison-Wesley.

Lefèvre, C. 2007. Discrete Compound Poisson Process with Curved Boundaries: Polynomial Structures and Recursions. *Methodology and Computing in Applied Probability, 9*(2), 243–262.

Lefèvre, C., S. Loisel. 2009. Finite-Time Ruin Probabilities for Discrete, Possibly Dependent, Claim Severities. *Methodology and Computing in Applied Probability, 11*(3), 425–441.

McNeil, A.J., R. Frey, P. Embrechts. 2005. *Quantitative Risk Management: Concepts, Techniques and Tools.* Princeton University Press, Princeton, NJ.

Panagiotelis, A., C. Czado, H. Joe. 2012. Pair Copula Constructions for Multivariate Discrete Data. *Journal of the American Statistical Association, 107*(499), 1063–1072.

Picard, P., C. Lefèvre. 1997. The probability of ruin in finite time with discrete claim size distribution. *Scandinavian Actuarial Journal, 1997*(1), 58–69.

Picard, P., C. Lefèvre. 2011. A new look at the homogeneous risk model. *Insurance: Mathematics and Economics, 49*(3), 512–519.

Sendova, K.P., R. Zitikis. 2012. The order statistic claim process with dependent claim frequencies and severities. *Journal of Statistical Theory and Practice, 6*, 597–620.

Vein, R., P. Dale. 1999. *Determinants and their applications in mathematical physics.* Springer-Verlag, New York.

Vitter, J. S. (1987) An efficient algorithm for sequential random sampling. *ACM Transactions on Mathematical Software, 13*(1), 58–67.

Table 1: Computation times (in sec) for $P(T > x)$ using formula (9) with the assumptions of Example 6.1 and parameter values $\lambda = 1, \alpha = 0.5, c = 1.25, x = 10$.

Appell polynomials computed by	$u = 0$	$u = 2$	$u = 4$	$u = 6$	$u = 8$	$u = 10$
Formula (14) – iteratively	1.23	5.65	13.04	28.94	80.18	282.55
Formula (14) – recursively	0.48	2.04	5.88	17.94	65.36	262.07
Formula (15)&(16)	0.47	1.84	5.29	17.04	64.10	261.40
Formula (15)&(17)&(16)	0.67	2.25	5.73	17.39	64.44	262.32

Table 2: Computation times (in sec) for $P(T > x)$ using formula (6) with the assumptions of Example 6.1 and parameter values $\lambda = 1, \alpha = 0.5, c = 1.25, x = 10$.

Appell polynomials computed by	$u = 0$	$u = 2$	$u = 4$	$u = 6$	$u = 8$	$u = 10$
Formula (14) – iteratively	0.70	3.09	7.60	18.77	58.14	218.29
Formula (14) – recursively	0.41	1.76	4.93	14.80	53.09	212.66
Formula (15)&(16)	0.42	1.61	4.46	13.99	52.24	212.68
Formula (15)&(17)&(16)	0.61	2.03	4.90	14.48	52.35	212.69

Table 3: Comparing the computational efficiency and accuracy of formulas (4) and (8) on the basis of Example 6.2 with parameter values $\lambda = 1, \alpha = 0.5, u = 1, x = 1$.

	Formula (4), $\epsilon = 10^{-6}$			Formula (4), $\epsilon = 10^{-4}$			Formula (8), $\epsilon = 10^{-4}$		
c	m_ϵ	$P(T > x)$	Time (sec)	m_ϵ	$P(T > x)$	Time (sec)	m_ϵ	$P(T > x)$	Time (sec)
1.0	5	0.599488	31.03	4	0.599481	9.80	8	0.599478	278.63
1.1	5	0.605719	30.56	4	0.605711	9.98	8	0.605705	282.63
1.2	5	0.611831	30.81	4	0.611821	11.06	8	0.611811	283.61
1.3	5	0.617826	33.07	4	0.617815	10.45	9	0.617823	678.39
1.4	5	0.623707	31.92	4	0.623693	9.87	9	0.623702	631.76
1.5	5	0.629475	32.87	4	0.629459	11.42	9	0.629468	646.83

Table 4: The truncating points, l_ϵ^* and m_ϵ^*, estimated using (19) with $k^* = 0$.

	λx	1		5		10		20		50		100	
ϵ	n	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*
10^{-4}	1	0	1	0	1	1	1	$> n$	–	$> n$	–	$> n$	–
	5	0	5	0	5	1	5	$> n$	–	$> n$	–	$> n$	–
	10	0	6	0	10	1	10	5	10	$> n$	–	$> n$	–
	20	0	6	0	15	1	20	5	20	$> n$	–	$> n$	–
	50	0	6	0	15	1	24	5	39	25	50	$> n$	–
	100	0	6	0	15	1	24	5	39	25	79	64	100
10^{-6}	1	0	1	0	1	0	1	$> n$	–	$> n$	–	$> n$	–
	5	0	5	0	5	0	5	3	5	$> n$	–	$> n$	–
	10	0	9	0	10	0	10	3	10	$> n$	–	$> n$	–
	20	0	9	0	19	0	20	3	20	20	20	$> n$	–
	50	0	9	0	19	0	28	3	45	20	50	$> n$	–
	100	0	9	0	19	0	28	3	45	20	88	55	100

Table 5: The truncating points, l_ϵ^* and m_ϵ^*, estimated using (18) with $k^* = 0$.

ϵ \ λx	1		5		10		20		50		100	
	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*	l_ϵ^*	m_ϵ^*
10^{-4}	0	6	0	15	1	24	5	39	25	79	64	141
10^{-6}	0	9	0	19	0	28	3	45	20	88	55	152

Table 6: The truncating point from above in (4), m_ϵ^*, estimated using (18) with varying k^* and $\epsilon = 10^{-3}$, under the assumptions of Example 6.2 with parameter values $\lambda = 1, \alpha = 0.5, u = 1, c = 1.25, x = 1$.

k^*	m_ϵ^*	$P(T > x)$	Time (sec)
0	5	0.614843	31.34
1	5	0.614843	31.34
2	4	0.614832	9.33
3	4	0.614832	9.33

Table 7: The truncating point from above in (4), m_ϵ^*, estimated using (18) with $k^* = 2$ and varying ϵ, under the assumptions of Example 6.2 with parameter values $\lambda = 1, \alpha = 0.5, u = 1, c = 1.25, x = 1$.

ϵ	m_ϵ^*	$P(T > x)$	Time (sec)
10^{-2}	3	0.614573	1.05
10^{-3}	4	0.614832	9.33
10^{-4}	6	0.614843	73.96
10^{-5}	7	0.614843	161.94
10^{-6}	8	0.614843	330.02

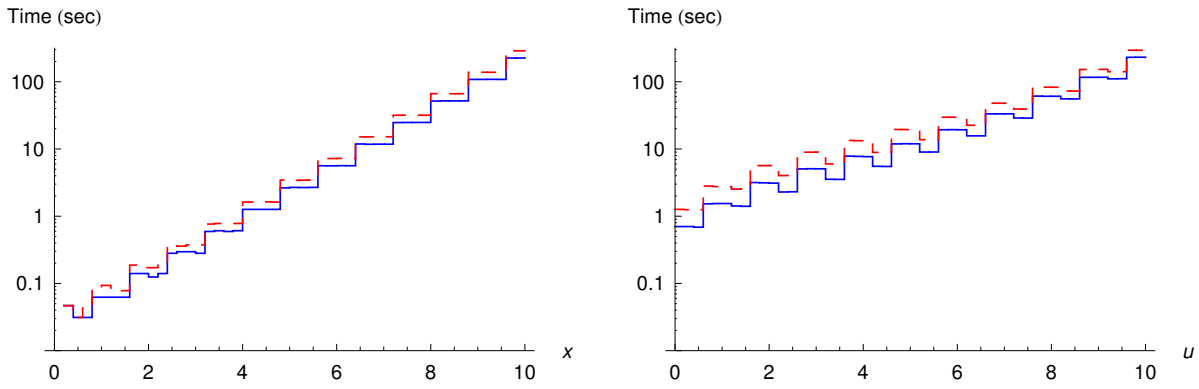
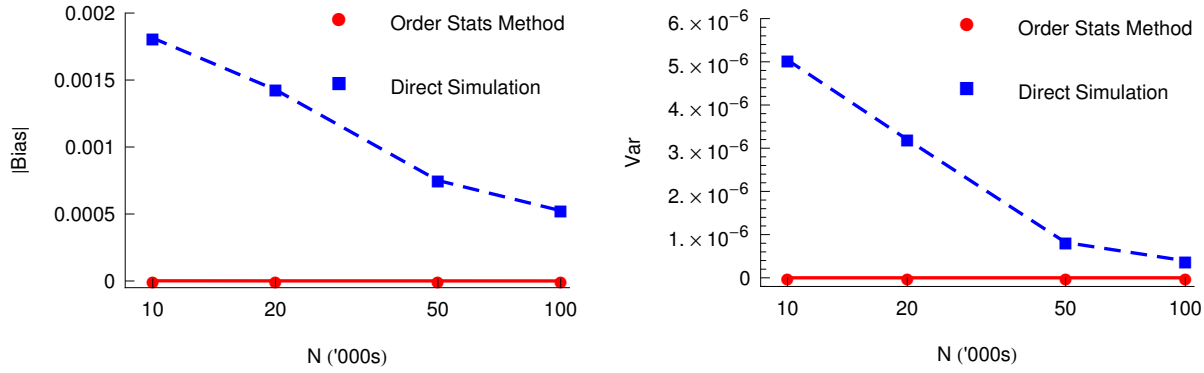
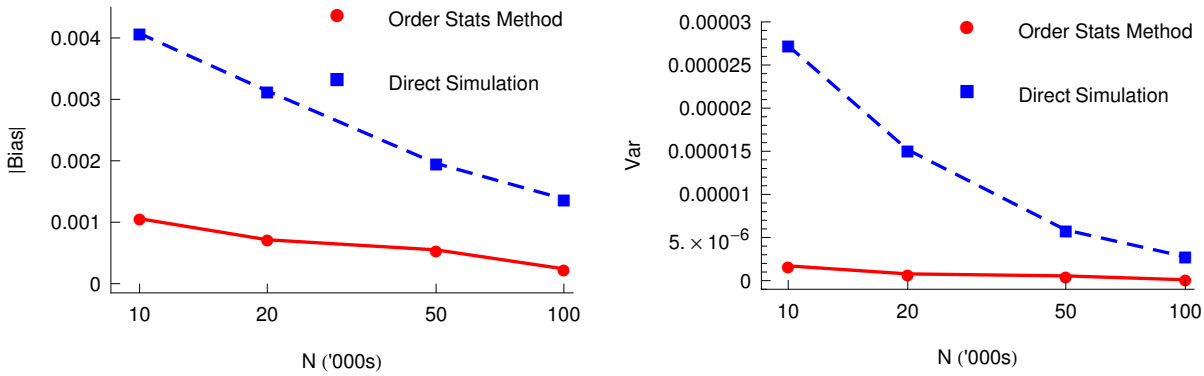


Figure 1: Computation times (in sec) for $P(T > x)$ using formula (6) - blue (solid) line, and formula (9) - red (dashed) line, with the assumptions of Example 6.1 and parameter values $\lambda = 1, \alpha = 0.5, c = 1.25$, left panel - $u = 10$, right panel - $x = 10$.

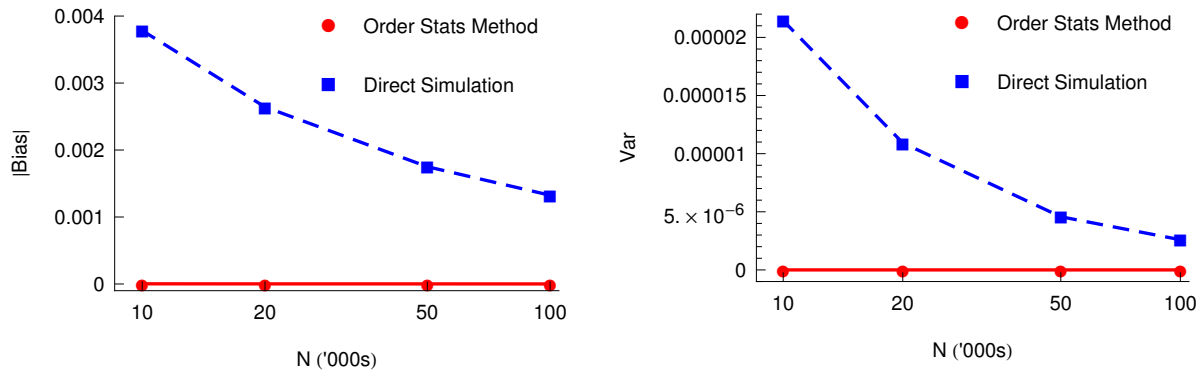


(a) λx=10, α=0.9, u=5, P(T>x)=0.0507

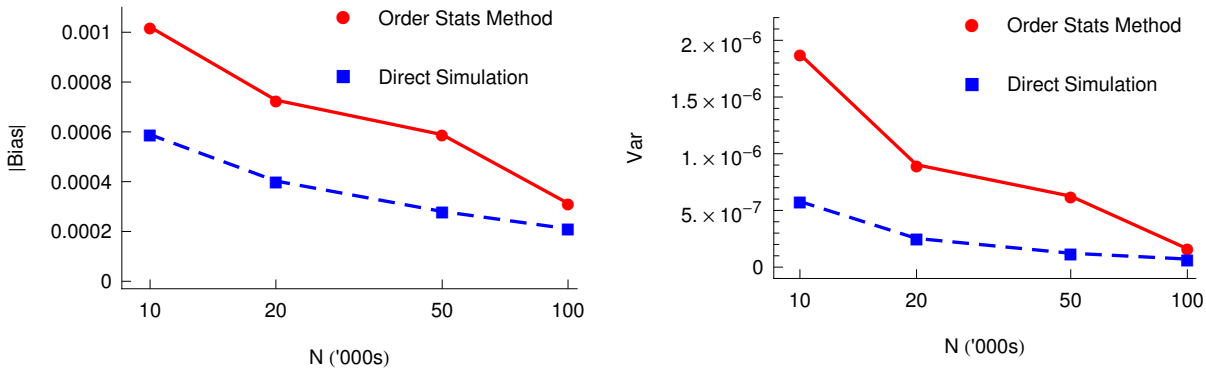


(b) λx=10, α=0.5, u=5, P(T>x)=0.4602

Figure 2: The absolute bias of the simulated estimates of $P(T > x)$ (left panels) and their corresponding variances (right panels) using both the order statistics simulation-based method (red solid line) and direct MC simulation (blue dashed line), obtained with the assumptions of Example 6.1. Parameter values: $\lambda = 1$, $c = 1$, $\epsilon = 10^{-4}$, $l_\epsilon^* = 0$, $m_\epsilon^* = 15$ ($n = 15$) and $M = 100$.

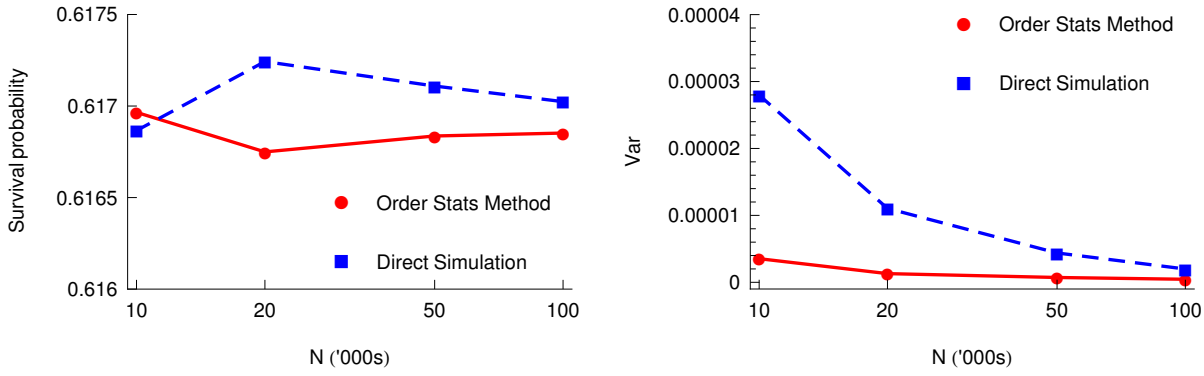


(a) λx=5, α=0.9, u=15, P(T>x)=0.5744

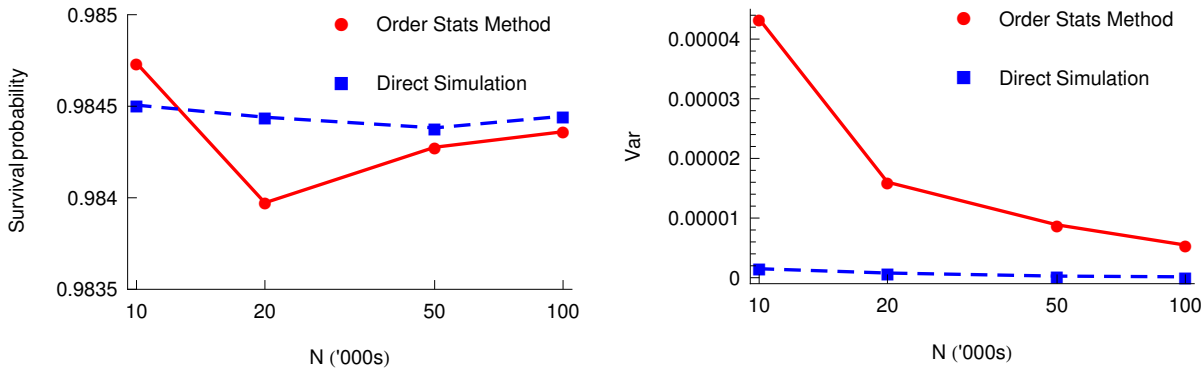


(b) λx=5, α=0.5, u=15, P(T>x)=0.9940

Figure 3: The absolute bias of the simulated estimates of $P(T > x)$ (left panels) and their corresponding variances (right panels) using both the order statistics simulation-based method (red solid line) and direct MC simulation (blue dashed line), obtained with the assumptions of Example 6.1. Parameter values: $\lambda = 1$, $c = 1$, $\epsilon = 10^{-4}$, $l_\epsilon^* = 0$, $m_\epsilon^* = 15$ ($n = 20$) and $M = 100$.

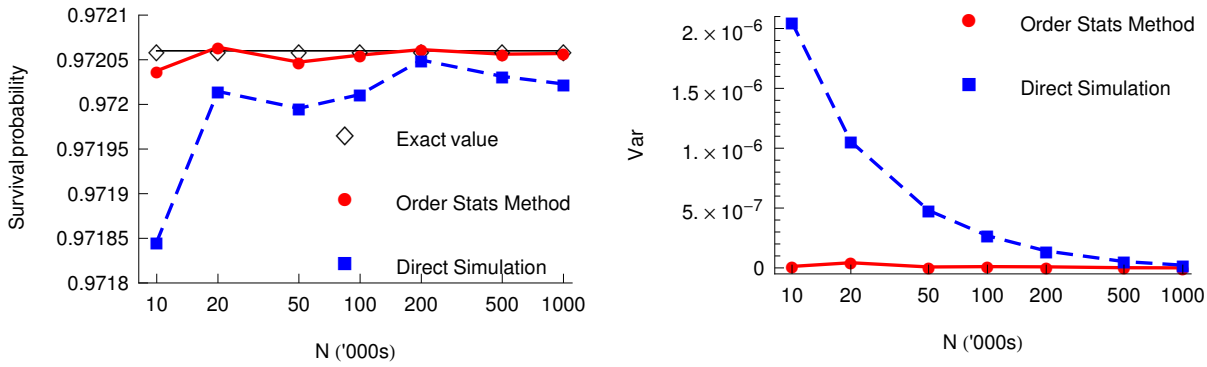


(a) λx=5, α=0.9, u=15

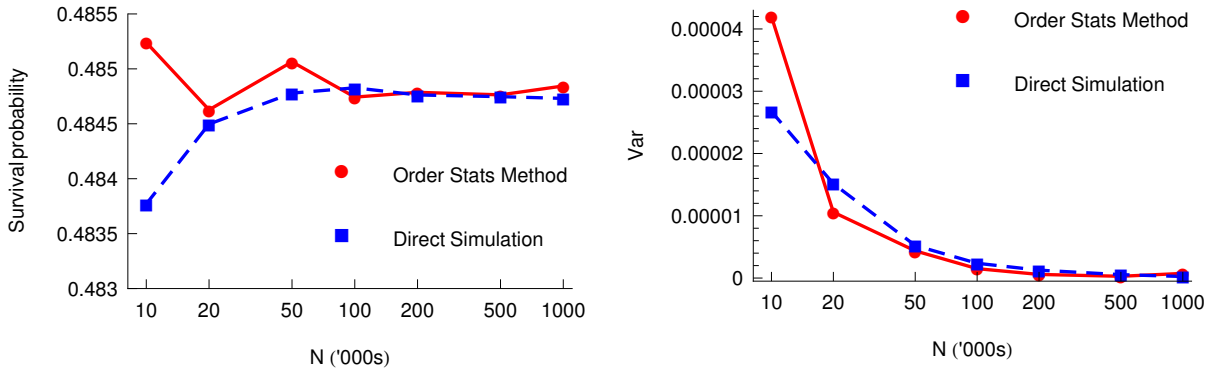


(b) λx=5, α=0.5, u=15

Figure 4: The simulated estimates of $P(T > x)$ (left panels) and their corresponding variances (right panels) using both the order statistics simulation-based method (red solid line) and direct MC simulation (blue dashed line), obtained with the assumptions of Example 6.3. Parameter values: $\lambda = 1$, $c = 1$, $\epsilon = 10^{-4}$, $l_\epsilon^* = 0$, $m_\epsilon^* = 15$ $(n = 20)$ and $M = 100$.

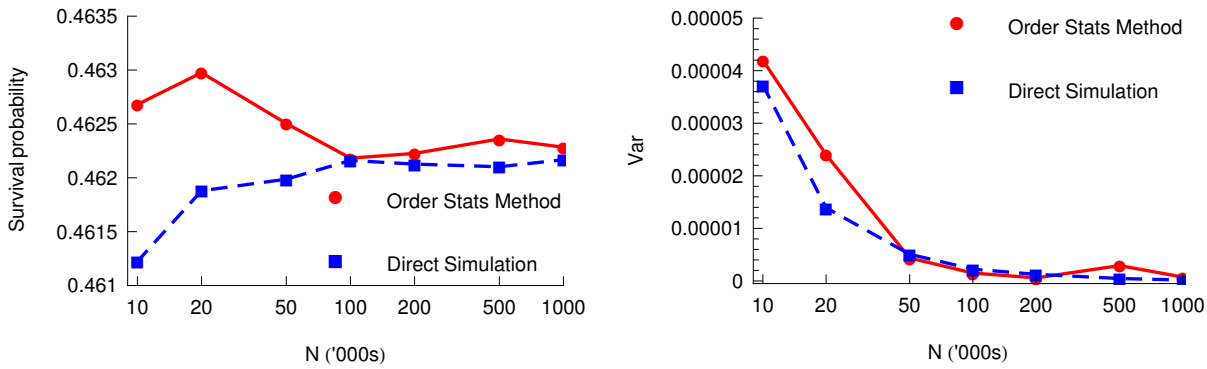


(a) λx=1, α_p=2, λ_p=2, θ=1, u=10, P(T>x)=0.972060



(b) λx=5, α_p=4, λ_p=2, θ=1, u=5

Figure 5: The simulated estimates of $P(T > x)$ (left panels) and their corresponding variances (right panels) using both the order statistics simulation-based method (red solid line) and direct MC simulation (blue dashed line), obtained with the assumptions of Example 6.4. Parameter values (a): $\lambda = 1$, $c = 3$, $\epsilon = 10^{-6}$, $l_\epsilon^* = 0$, $m_\epsilon^* = 9$ and $M = 100$. Parameter values (b): $\lambda = 1$, $c = 1$, $\epsilon = 10^{-4}$, $l_\epsilon^* = 0$, $m_\epsilon^* = 15$ and $M = 100$.



(a) λx=5, α_p=4, λ_p=2, θ=1, u=4

Figure 6: The simulated estimates of $P(T > x)$ (left panel) and their corresponding variances (right panel) using both the order statistics simulation-based method (red solid line) and direct MC simulation (blue dashed line), obtained with the assumptions of Example 6.5. Parameter values: $\lambda = 1$, $c = 1$, $\epsilon = 10^{-4}$, $l_\epsilon^* = 0$, $m_\epsilon^* = 15$ and $M = 100$.